พันธนีย์ กำแพงทอง

Developer & Workshop

VB Debugging หน่วยลาดตระเวนในการเขียนโปรแกรม Visual Basic ตอน ล้วงความลับการดีบักค่าตัวแปร

โปรแกรมวิชวลเบสิกมีตัวช่วยในการเขียนโปรแกรมหนึ่งในนั้นต้องมีเครื่องมือเพื่อให้โปรแกรมที่ออกมานั้นสามารถทำงานได้ถูกต้อง มีความรวดเร็วและมีประสิทธิภาพมากที่สุด

Visual Basic Programming

จากบทความในไมโครคอมพิวเตอร์ ฉบับที่ 192 ผู้เขียนได้นำเสนอบทความให้ผู้อ่านได้รู้จักกับเทคนิคการดีบักโปรแกรมวิชวลเบสิกกันไปแล้ว มีเทคนิคหนึ่งที่เรียกว่า "Auto Data Tip" สำหรับเรียกดูค่าตัวแปรต่าง ๆ ที่กำหนดไว้ในโปรแกรม ซึ่งจะทำให้การดีบักโปรแกรมทำได้อย่างรวดเร็วและมีประสิทธิภาพ อันที่จริงแล้ววิชวลเบสิกยังมีวิธีดีบักค่าตัวแปรอีกหลายวิธี คราวนี้ผู้เขียนจึงนำมาแนะนำให้ท่านได้รู้จักกัน

เทคนิคการดีบักค่าตัวแปรในโปรแกรมวิชวลเบสิก

นอกจากวิธี "Auto Data Tip" แล้วท่านยังสามารถตรวจสอบค่าต่าง ๆ ขณะทำการรันโปรแกรมได้โดยใช้เครื่องมือที่วิชวลเบสิกเตรียมไว้ให้ดังนี้

- Call Stack
- Locals Window
- Watch Window / Quick Watch
- Immediate Window

การใช้งาน Call Stack

ท่านสามารถเรียกใช้ Call Stack ได้จากการคลิกที่เมนู [View] เลือกหัวข้อ [Call Stack] หรือกด Ctrl-L หน้าจอนี้จะแสดงลำดับของ procedure ที่ทำงานอยู่ในรูปแบบของ stack คือ FILO (First In Last Out) หมายถึง procedure ที่ถูกเรียกใช้ก่อนจะแสดงให้เห็นในลำดับล่างสุด ดังรูปที่ 1

ตัวอย่างในรูปที่ 1 ขณะนี้โปรแกรมกำลังเรียกใช้ฟังก์ชัน fnSearch__Data จากที่ท่านเห็นในบรรทัดที่ 1 "Project1.AsfrmProc001.fnSearch__Data" ฟังก์ชันนี้ถูกเรียกมาจาก Event KeyDown ของ Textbox ชื่อ txtAsset__Name ส่วนในบรรทัดที่ 2 "Project.AsfrmProc001.txtAsset__Name__KeyDown"

รูปที่ 1 แสดงหน้าจอ Call Stack

เมื่อต้องการทราบแหล่งที่มาของแต่ละ procedure ให้คลิกเลือก procedure ที่ต้องการ แล้วคลิกที่ปุ่ม Show จะแสดงลูกศรทางด้านซ้ายของจอตรงที่คำสั่งดังรูปที่ 2

ท่านต้องกำหนด Break Point จึงจะสามารถเรียกหน้าจอ Call Stack ได้ ซึ่งหน้าจอ Call Stack จะทำให้ทราบว่า procedure ใดบ้างทำงานอยู่ในขณะนั้น และอาจช่วยตรวจสอบได้ว่าท่านเขียนโปรแกรมที่มีการ Call procedure มากเกินไปหรือไม่ หรืออาจจะทำให้หมดข้อสงสัยว่าโปรแกรมของท่านวิ่งเข้าไปใน procedure นั้นได้อย่างไรทั้ง ๆ ที่ไม่ต้องการให้เป็นอย่างนั้น พิจารณาตัวอย่างจากรูปที่ 3 เพื่อสร้างความเข้าใจยิ่งขึ้น

จากรูปที่ 3 เมื่อคลิกปุ่ม (Command Button) cmdCallStack จะทำการเรียก subProcedure__1 แล้ว subProcedure__1 ก็จะเรียก subProcedure__2 จากนั้น subProcedure__2 ก็จะเรียก subProcedure__3 ตามลำดับ ดังนั้นเมื่อท่านกำหนด Break point ที่ subProcedure__3 แล้วเรียกหน้าจอ Call Stack ขึ้นมาตรวจสอบจะปรากฏดังรูปที่ 4

การใช้งาน Locals Window

ท่านสามารถเรียกใช้งาน Locals Window ได้โดยคลิกที่เมนู [View] หลังจากนั้นเลือกหัวข้อ [Locals Window] จะปรากฏหน้าจอแสดงค่าของตัวแปรใน procedure หรือ function ที่กำลังทำงานในปัจจุบัน ขณะที่รันโปรแกรมผ่านฟังก์ชันถ้าท่านต้องการทราบค่าตัวแปรหลายตัวในฟังก์ชันพร้อม ๆ กันสามารถตรวจสอบได้โดยไม่ต้องเลื่อนเมาส์ไปที่ตัวแปรทีละตัวให้เสียเวลา ผู้อ่านควรทำความเข้าใจกับการเขียนโค้ดดังรูปที่ 5 ด้วย

ตัวอย่างในรูปที่ 5 เป็นการเขียนโปรแกรมเพื่อนับจำนวนคำที่ระบุในประโยคที่กำหนด เมื่อกำหนด Break point ที่จุดใดจุดหนึ่งใน procedure cmdCountWord__Click() และเรียก Locals Window ขึ้นมาตรวจสอบ ท่านก็จะทราบค่าของตัวแปร intPosition และ intCounter ในฟังก์ชันดังรูปที่ 6

ใช้งาน Watch Window / Quick Watch

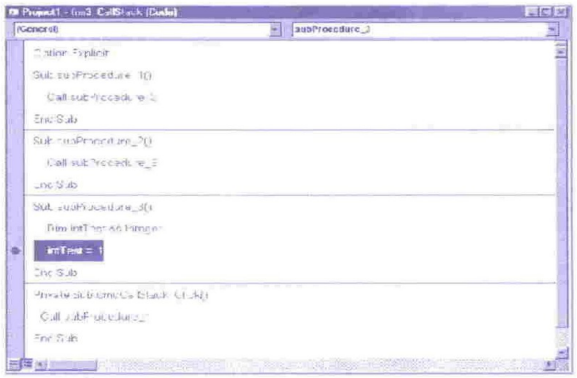
Watch Window ใช้ในการตรวจสอบข้อมูลหรือค่าตัวแปรขณะทำการรันโปรแกรม โดยมีฟังก์ชันการทำงานให้ท่านเลือกใช้แบ่งได้ 3 รูปแบบดังนี้ (พิจารณาจาก Watch Type) ดังรูปที่ 7

แบบที่ 1 ตรวจค่าของข้อมูลหรือตัวแปร (Watch Expression)

แบบที่ 2 กำหนดให้หยุดการทำงานเมื่อตัวแปรมีค่าตรงกับค่าที่กำหนด (Break When Value Is True)

แบบที่ 3 กำหนดให้หยุดทำงานเมื่อตัวแปรปรับเปลี่ยนค่า (Break When Value Changes)

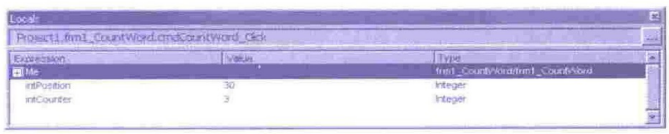
รูปที่ 2 แสดงหน้าจอที่มาของโปรเซสใน Call Stack



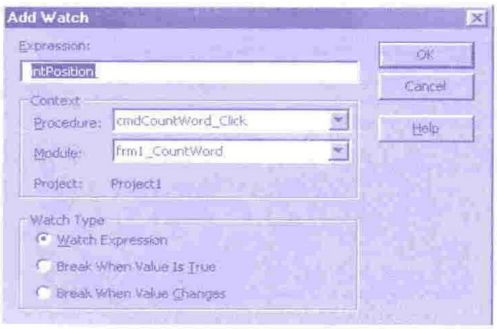
รูปที่ 3 แสดงหน้าจอการเขียนโปรแกรม ในลักษณะการ Call Procedure

รูปที่ 4 แสดงหน้าจอ Call Stack จากการ Coding ในรูปที่ 3

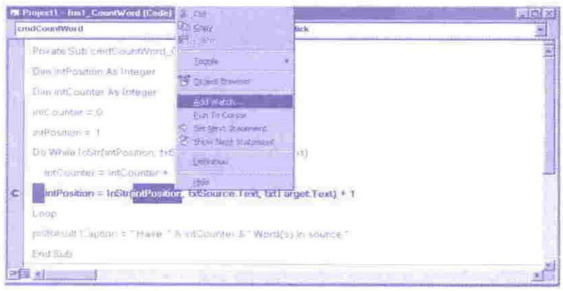
รูปที่ 5 แสดงหน้าจอการCoding เพื่อนับจำนวนคำในประโยค



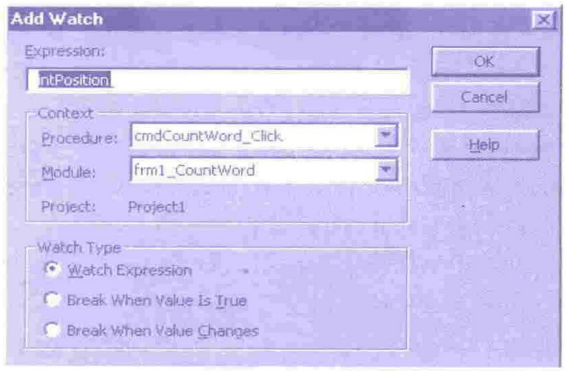
รูปที่ 6 แสดงหน้าจอ Locals Window



รูปที่ 7 แสดงหน้าจอ Add Watch

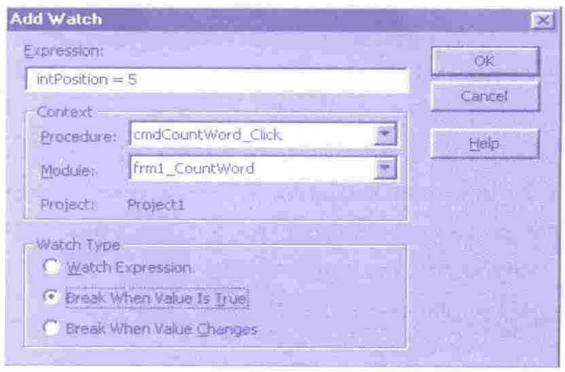


รูปที่ 8 แสดงวิธีการเรียกหน้าจอ Add Watch

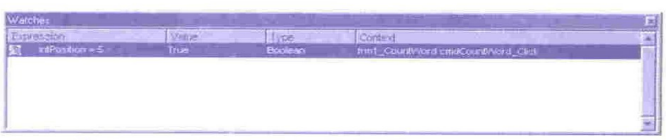


รูปที่ 9 แสดงหน้าจอ Add Watch ของตัวแปร intPosition (แบบที่ 1)

รูปที่ 10 แสดงค่าของตัวแปรในหน้าจอ Watches



รูปที่ 11 แสดงการกำหนดเงื่อนไขในการตรวจสอบค่าของตัวแปรในหน้าจอ Add Watch (แบบที่ 2)



รูปที่ 12 แสดงค่าของตัวแปร intPosition เมื่อเงื่อนไขที่กำหนดเป็นจริง

ขั้นตอนการทำงานแบบที่ 1 (Watch Expression)

1. เลือกตัวแปรหรือข้อมูลที่ต้องการตรวจสอบค่า หลังจากนั้นคลิกขวาเลือกหัวข้อ Add Watch ดังรูปที่ 8

2. ในหัวข้อ Watch Type เลือก Watch Expression จากนั้นคลิกปุ่ม OK จะปรากฏหน้าจอแสดงค่าของตัวแปรดังรูปที่ 9

3. หลังจากนั้นให้คลิกที่ปุ่ม OK จะปรากฏหน้าจอดังรูปที่ 10 และมองเห็นค่าของตัวแปร intPosition

ขั้นตอนการทำงานแบบที่ 2 (Break When Value Is True)

ขั้นตอนการทำงานเหมือนกับแบบที่ 1 แต่ในหัวข้อที่ 2 ให้เลือก Watch Type เป็น Break When Value Is True ส่วน Expression ด้านบนสามารถกำหนดเป็นเงื่อนไขต่าง ๆ เช่น intPosition = 5 ซึ่งท่านสามารถกำหนดได้ก่อนทำการรันโปรแกรมดังรูปที่ 11

จากตัวอย่างโปรแกรมจะหยุดการทำงานชั่วคราวทันทีที่ intPosition = 5 เป็นจริง (True)โดยไม่ต้องกำหนด Break point การทำงานในลักษณะนี้จะช่วยให้ท่านทราบได้ทันทีว่า ตัวแปรของท่านมีค่าตามที่ต้องการหรือไม่ หรือเมื่อตัวแปรมีค่าตามที่กำหนดไว้จะมีผลอย่างไรกับการทำงานของโปรแกรม เช่น อาจจะตรวจสอบค่าของตัวแปรตัวอื่น ๆ ว่าถูกต้องหรือไม่ โดยที่ผู้อ่านไม่ต้องหยุดโปรแกรม และกด F5 เพื่อรันผ่านตัวแปรไปเรื่อย ๆ จนถึงค่าที่ต้องการ จะช่วยประหยัดเวลาในการตรวจสอบได้ดีทีเดียว โดยเฉพาะในกรณีที่ค่าตัวแปรที่ทดสอบมีค่าที่เป็นไปได้หลาย ๆ ค่า คงจะลำบากมิใช่น้อยถ้าท่านจะต้องหยุดโปรแกรมและกด F5 ไปเรื่อย ๆ จนกระทั่งตัวแปรมีค่าตามที่ท่านต้องการ ยิ่งถ้าไม่ทราบลำดับของค่าภายในตัวแปรที่จะเกิดขึ้นในแต่ละครั้งที่ท่านกด F5 แล้วก็ไม่รู้ว่าจะต้องกด F5 กี่ครั้งจึงจะตรวจสอบค่าต่าง ๆ ได้ สำหรับผลลัพธ์จากการตรวจสอบวิธีนี้จะปรากฏดังรูปที่ 12

ขั้นตอนการทำงานแบบที่ 3 (Break When Value Changes)

ขั้นตอนการทำงานเหมือนกับแบบที่ 1 แต่ในหัวข้อที่ 2 ให้เลือก Watch Type เป็น Break When Value Changes ส่วน Expression ด้านบนกำหนดชื่อตัวแปรที่ต้องการตรวจสอบค่าดังรูปที่ 13 กำหนดเป็นตัวแปร intPosition ซึ่งสามารถกำหนดได้ก่อนรันโปรแกรม

จากตัวอย่างในรูปที่ 13 เมื่อทำการรันโปรแกรมทันทีที่ตัวแปร intPosition มีการเปลี่ยนแปลงค่าโปรแกรมจะหยุดทำงานชั่วคราวและแสดงหน้าจอ Watches ให้เห็นค่าของตัวแปร intPosition

หลังจากนั้นเมื่อท่านกด F5 เพื่อรันโปรแกรมต่อ โปรแกรมก็จะทำงานต่อไปเรื่อย ๆ และจะหยุดทำงานอีกครั้งเมื่อ intPosition เปลี่ยนแปลงค่าไปจากเดิม

การใช้งาน Quick Watch

หน้าจอ Quick Watch เป็นหน้าจอที่ใช้ในการตรวจสอบค่าของตัวแปรทีละตัว ท่านสามารถตรวจสอบค่าของตัวแปรภายในภายในโปรแกรมผ่านหน้าจอ Quick Watch ได้ตามวิธีการดังนี้

1. เลือกตัวแปรที่ต้องการตรวจสอบค่าโดยดับเบิลคลิกที่ตัวแปรเพื่อให้เกิดแถบสว่างหรือให้เคอร์เซอร์อยู่ที่ตัวแปรนั้น

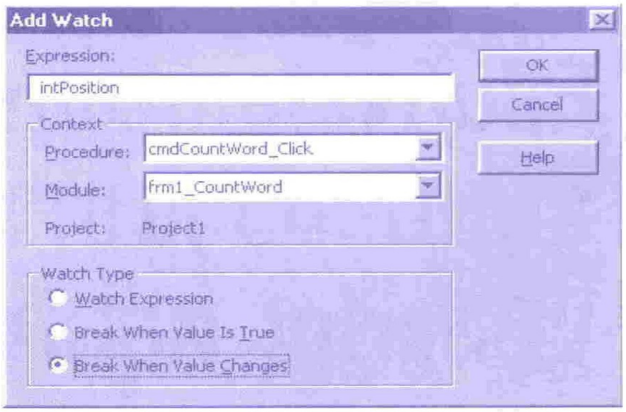
2. คลิกที่เมนู [Debug] และเลือกหัวข้อ [Quick Watch] หรือกด Shift + F9 จะปรากฏหน้าจอดังรูปที่ 14

จากตัวอย่างในรูปที่ 14 ถ้าคลิกที่ปุ่ม Add ตัวแปร intPosition จะถูกเพิ่มเข้าไปในหน้าจอ Watch Window ทันที การทำงานในลักษณะนี้เหมาะสำหรับกรณีที่ต้องการให้ตัวแปร intPosition ถูกแสดงในหน้าจอ Watches อย่างรวดเร็วโดยไม่ผ่านหน้าจอ Add Watch เพื่อกำหนดเงื่อนไขพิเศษอื่น ๆ เพิ่มเติม ส่วนใหญ่มักไม่ค่อยได้ใช้งาน Quick Watch กันเท่าไร แต่จะนิยมใช้วิธีการ Add Watch ลงในหน้าจอ Watches โดยตรงมากกว่า และปกติถ้าเลือกออปชัน Auto Data Tips ที่ได้กล่าวไว้แล้วในตอนต้น เมื่อใช้เมาส์เลื่อนไปในตำแหน่งของตัวแปรก็จะปรากฏค่าตัวแปรให้ทราบได้ทันที โดยไม่ต้องเรียกหน้าจอ Quick Watch

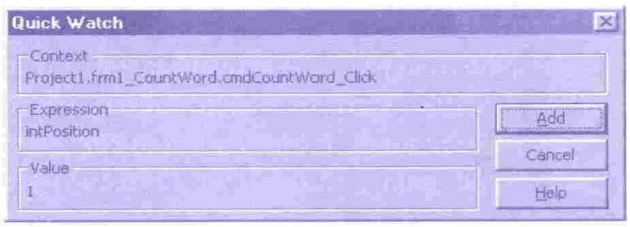
การใช้งาน Immediate Windows

การเรียกใช้งาน Immediate Windows ทำได้โดยคลิกเมนู [View] เลือกหัวข้อ [Immediate Windows] หรือกด Ctrl+G หน้าจอ Immediate Window นอกจากจะใช้ในการแสดงค่าจากคำสั่ง Debug Print แล้วยังใช้ในการตรวจสอบค่าของตัวแปรหรือคำสั่งต่าง ๆ เมื่อโปรแกรมหยุดการทำงานชั่วขณะจากการเบรคโปรแกรมของท่าน หน้าจอ Immediate จะทำหน้าที่คล้าย ๆ กับกระดาษทด เมื่ออยากจะทราบค่าอะไรก็พิมพ์ป้อนเข้าไปในหน้าจอ Immediate Windows โดยใช้คำสั่ง Print ดังรูปที่ 15

ตัวอย่างในรูปที่ 15 เป็นการตรวจสอบค่าของ txtSource.Text ซึ่งเป็นออบเจ็กต์ Textbox แสดงให้เห็นว่า txtSource.Text เก็บค่าเป็นข้อความ "Testing Immediate Window" หรืออาจจะใช้เครื่องหมาย "?" แทนคำสั่ง Print ก็ได้



รูปที่ 13 แสดงการเช็คค่าของตัวแปรในหน้าจอ Add Watch (แบบที่ 3)



รูปที่ 14 แสดงหน้าจอ Quick Watch ของตัวแปร intPosition

รูปที่ 15 แสดงหน้าจอ Immediate

ก่อนจบตอน

ผู้เขียนได้แนะนำวิธีการต่าง ๆ เพื่อช่วยในการดีบักโปรแกรมไปแล้ว ทีนี้ลองเลือกใช้ตามแต่ถนัดก็แล้วกัน วิธีไหนก็ได้ขอให้แก้โปรแกรมผ่าน นอกจากวิธีการต่าง ๆ ที่ได้อธิบายไปทั้งหมดแล้ว ถ้าจะให้ดีท่านต้องรู้จักวิธีรันโปรแกรมแบบต่าง ๆ เพื่อให้สามารถตรวจสอบข้อผิดพลาดในตำแหน่งที่ต้องการได้อย่างรวดเร็ว และต้องรู้จักควบคุมข้อผิดพลาดต่าง ๆ ที่อาจเกิดขึ้นได้ในขณะที่ผู้ใช้เรียกใช้งานโปรแกรมของท่าน เช่น กรณีกดปุ่มบันทึกไฟล์ลงในไดรฟ์ A: แต่ลืมใส่แผ่นดิสเกตต์ที่ไดรฟ์ A: ถ้าไม่ได้เขียนคำสั่งควบคุมไว้ในโปรแกรม จะปรากฏข้อความแสดงให้ทราบว่าเกิดความผิดพลาดขึ้น ผู้ใช้อาจสงสัยว่าเกิดอะไรขึ้น

แล้วพบกันใหม่ในฉบับหน้า สวัสดีค่ะ